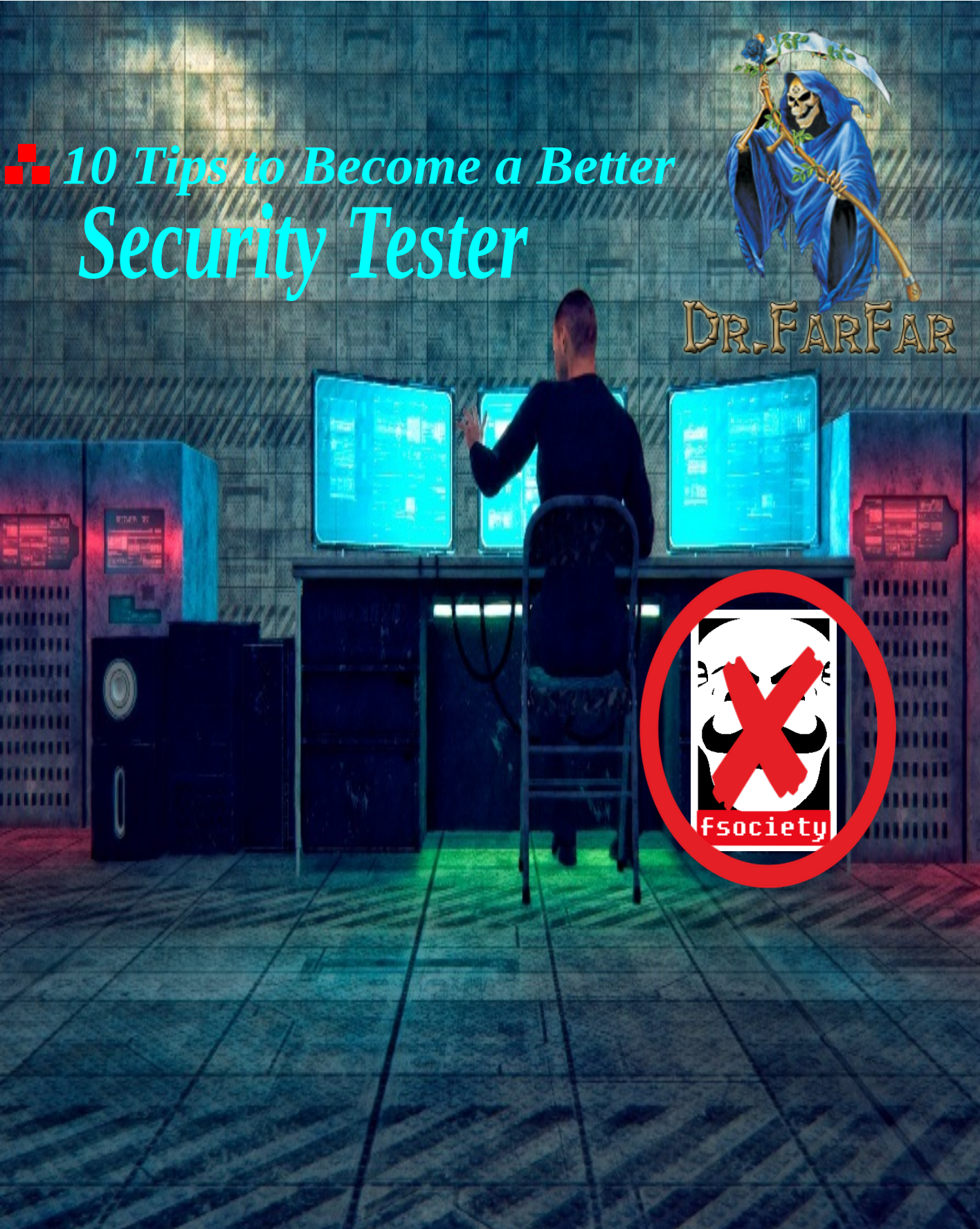
10 Tips to Become a Better
Security Tester
Dr.FarFar
fsociety

Ten Tips to Become a Better Security Tester

Being a good security tester takes a lot of passion and dedication. An interesting side to the security tester profession is that there will always be new threats and vulnerabilities to dig into. While the never-ending flood of reported security breaches may seem somewhat intimidating, the following list of tips on how to become a better security tester might help you stay afloat.

1. Learn How to Program

First off: you don't need to be an expert programmer to be a good security tester. In fact, you don't need to know anything at all about programming when you first start out. But as you slowly but surely get better at finding and properly reporting security weaknesses, you will probably realize that having at least decent programming skills is a great advantage.

Knowing how to program, or at least being able to do minor modifications to someone else's programming code, will give you a better understanding of how vulnerabilities in software can be exploited.

An example would be proof-of-concept code released to demonstrate a weakness in a specific application. In some cases, the proof-of-concept code will work fine without any modification. But in many cases, the proof-of-concept code won't execute properly in your environment without you modifying the code in some way. The difficulty of successfully modifying proof-of-concept code lies surprisingly often in trying to understand what the original developer was trying to do and in what way. The world of computers would be a much better place if every one of us put some effort into commenting our source code (but that's a different book).

Learning to be a good programmer takes years. But learning to program well enough to get some useful work done can be accomplished in a few weeks. A fabulous introduction to programming for complete beginners is the *Learn Code the Hard Way* series developed by Zed A. Shaw, found at http://learncodethehardway.org/.

A small example of useful code that you can learn to write within a few weeks of studying could be the following simple application. It's a Python script that footprints web servers in a very elementary way.

```
#import the Requests HTTP library
import requests
'''
send an HTTP request to www.artandhacks.se and store the response
in an object called req
'''
req = requests.get('http://www.artandhacks.se')
```

```
#print the returned HTML code
print("Returned HTML code:")
print(req.content)
print ("\n")

#print the returned HTTP headers
print("Returned headers:\n")
print(req.headers)
```

When executed, the script generates the following output:

```
python webserverEnum.py
Returned HTML code:
<html>
<head>
<title>ART&HACKS</title>
</head>
<body>
<center>
<br/>
<img src="logo.svg"/>
</center>
</body>
</html>

Returned HTTP headers:

{'content-length': '104', 'via': '1.1 varnish', 'content-encoding': 'gzip', 'accept-ranges':
'bytes', 'vary': 'Accept-Encoding', 'server': 'Apache', 'last-modified': 'Sat, 17 Nov 2012
11:47:58 GMT', 'connection': 'keep-alive', 'x-varnish': '1818156263', 'etag': '"238c50ae-
86-4ceaf74a922ec"', 'date': 'Wed, 27 Jan 2016 10:50:03 GMT', 'content-type': 'text/html',
'age': '0'}
```

2. It's Elementary, Watson

Have you been to too many security presentations where phrases like *securing the cloud*, *advanced persistent threats*, and *next generation X* flew across the room like futuristic tongue twisters of digital Armageddon? I sure have. As much as I love to research a newly emerged malware kit, or try out the new hack tool of the day, I always try to find the fundamental building blocks of whatever it is I have in front of me. Because while technology has changed the way many of us live our lives, the fundamentals of computing still work the same way that it has for a long time. That's why it's always a good idea to brush up on your TCP/IP skills, to arm yourself with a basic understanding of cryptography and to know more than a handful of UNIX commands. Having a fundamental understanding of how computers work and communicate will make it a lot easier to learn new security concepts, programming languages, network security devices, and so forth.

So pay a visit to the local computer museum, ask the man or woman who operated your bank's mainframe in 1964 a million questions, dive into the 1978 classic *The C Programming Language* by Brian Kernighan and Dennis Ritchie. Because regardless of smart marketing and hype X, data is still just data and a threat is still just a threat.

3. Read *The Boy Who Cried Wolf*

The story has undergone many transformations over the years, but any version of *The Boy Who Cried Wolf* should be a mandatory read for any security tester. When you break into computers systems for a living, it is all too easy to get the idea that every vulnerability of every system is a recipe for disaster. And that all the hacker wolves out there are just waiting to sink their teeth into whatever kind of digital meat that people are trying to protect.

Yes - it's of course true that a web server that is leaking passwords all over the Internet should be taken offline and reconfigured immediately. However, if the web server in question doesn't contain any sensitive information, then quickly addressing such vulnerability probably isn't all that critical.

My humble opinion is that a professional security tester should hold the wolf crying back until she uncovers vulnerability that hacking carnivores of the Internet can use to actually get a hold of sensitive data. On the other hand, if you're up against state-sponsored actors then you should probably cry wolf a lot more than you already might do: http://www.wired.com/2016/01/nsa-hacker-chief-explains-how-to-keep-him-out-of-your-system/.

4. Read Read Read Write Write Write

I read it from cover to cover. Tried all the hacks. Then I would start all over again. The first edition of *Hacking Exposed*[1] was the book that made want to be a hacker. In all honesty, I had a very vague idea of what a hacker actually was. But I didn't let such a fundamental piece of my future career puzzle hold me back. So I read and I hacked. Hacked and read. I felt as if I was armed to the teeth with knowledge, and that the world was simply waiting for me to do the best security tests known to man.

But it didn't take long before I realized that you can be the world's best hacker and still carve out some pretty terrible security tests. The reason is that you need to master two things equally well if you want to be at least a decent security tester: hacking *and* writing.

Learning to master any of the two crafts is most likely a lifelong journey. So what I believe you should do it to get right down to business and start reading as often as you can. Read poorly plotted crime novels, read Kafka, read Zagajewski. Just keep on reading.

Remember that the big difference between a hacker and a security tester is that the latter must hand in a report when she's done. Only good writers can write good reports. So perfect your craft by writing blog posts, poetry, short stories, a new testament, or anything else that makes you think about how to present a story and how to structure text. For me, writing this book was ridiculously difficult, but I believe my habit of reading a lot made it somewhat easier.

5. Learn to Spot the Shape that Breaks the Pattern

A couple of years ago, I did a security test for a multinational company specializing in headhunting skilled workers for well-paid jobs in the biotech industry. The main focus of the security test was the company's website, where job-seekers would upload their resumes hoping to get that crucial first interview.

The security of the website was unusually good. The only issue I found was a SSL/TLS-certificate that was about to expire within the coming week. In short, the company had little to worry about when it came to their most valuable digital asset.

However, during a network scan of their entire network I discovered a web server that was providing its visitors with some unexpected content. I never paid much attention to my French teacher in high school, but it was obvious that the web server I had found was serving pages in French that seemed to have little to do with the company's regular dealings.

---

[1] http://www.hackingexposed.com/

It turned out that an employee had put up his personal website on one of the company's Internet-connected networks. The web server, and its served content, had no apparent security issues. So running the discovery of this unexpected server through the CVSS model, and coming out with a value on the other side, wouldn't have rung any alarm bells. The CVSS value would just have flown below our radar.

Trying to determine what is appropriate, and what isn't, is a highly subjective operation. A completely acceptable configuration found on one network would be out of the question on someone else's network.

In this scenario, automated use of a vulnerability scanner like *Nessus* probably wouldn't have been enough to uncover the inappropriate web server. An automated scanner would most likely have rated the server as secure, only to move on further down the network.

Learning how to spot the shape that breaks the pattern is never easy - but it's a splendid skill to have.

6. Put Your Money where Your Mouth is (Most of the Time)

Guilty your honor. I plead guilty to having talked about a security issue without knowing much about it. I guess we all take shortcuts now and then. The sheer amount of vulnerabilities uncovered each day is enough to make any security professional reconsider her career choice. As much as we love to find out everything there is to know about a specific threat, there is often not enough hours in a day to always do so. This is even truer if you have something that at least remotely resembles a life outside the world of computers. But it is also true that the IT security industry needs more people who know what they are talking about.

After a few minutes of research, anyone can hold a presentation on the importance of properly salting hashes with unique values - or over some other "best practice." Don't get me wrong - such a presentation can be great for everyone involved. However, good security testers can't be all talk - they also need to walk the walk.

So before you lecture a poor soul over his poorly designed hashing process, design and build one yourself. I'm confident that you will have to make more security-related compromises when you're actually implementing your idea, than you did when your Einstein-like recommendation was still on the drawing board.

7. Tap Into the Noise

I've already brought up the idea that a good security tester should read and write a lot. If you find the idea of writing anything but work-related reports dreadful, and the idea of turning reading into a hobby even worse - you should at least make an effort to learn where you can catch up on the latest IT security news.

At the time of this writing, the websites listed below are a few of my favorite resources for security-related matters.

- IT security news: `https://itsecuritything.com/`, `https://nakedsecurity.sophos.com/` and `http://thehackernews.com/`
- Malware news and analysis: `http://malware.dontneedcoffee.com/` and `https://malwr.com/`
- Industry approved gurus: `http://krebsonsecurity.com/` and `https://www.schneier.com/`

The point I'm trying to make is not that *it's next to impossible to be a good security tester without keeping up with what's going on*. I'm the first person to admit that trying to absorb, and remember, everything that's happening out there can make anyone feel like they have amnesia. However, not keeping an eye on the unfolding of the latest and greatest stories out there is simply not an alternative.

8. Watch the Movie *Wargames*

In the 1983 classic *Wargames*, David Lightman mistakenly breaks into the *North American Aerospace Defense Command* using his dial-up connection and his home computer. His computer break-in triggers a chain of events that brings the world close to a third world war.

Any contemporary security tester will tell you that the vulnerabilities that David accidentally exploited in the movie can still be found today. The vulnerabilities include poor account management, a too generous remote connection policy, and a poor network segmentation.

So how will watching a movie make you a better security tester? I believe it's because it tells a remarkably good story on the importance of IT security. And also because it was a story about IT security that reached beyond the small world of security enthusiasts. If the moviemakers could use something as dry as IT security to tell a fascinating story - then so can you when you're writing up that final security test report. You may not be able to repeat the movie's success at the box office, but it should inspire you to write a good report for your next security test.

And yeah - the movie will give you a historical understanding of where the terms "war-dialing" and its newer cousin "war-driving" come from.

9. Know that Old Vulnerabilities Never Get Old

I have tried to not get surprised every time I stumble upon a vulnerability that "should not be there." The simple truth is that I would be a rather wealthy security tester if I had been given a dollar for every time I learned that default credentials could be used to gain root access to a system within seconds.

It's easy to believe that a security tester can only gain access to sensitive data by exploiting newly discovered vulnerabilities. The idea that someone could take control of a system, or an entire network, by taking advantage of a ten-year-old exploit seems absurd. The somewhat sad state of affairs is that such an opportunity exists all too often.

Take a newly developed web application, for example: an organization can spend a lot of time and resources on building a secure and well-functioning application, but neglect to remember to change the default administrator password. Building a secure house, while leaving the front door unlocked, is a rather pointless exercise.

A good way to approach any system is to assume that it is as insecure as one can possibly imagine - that way you will never forget to test the easy stuff first. You'll be surprised how often it can give you that dollar.

10. Have Fun

Any security test will suffer from the security tester not having any fun. I've had many different jobs in IT. I've gotten yelled at over the phone as first-line support, I've deployed thousands of servers as a system administrator, and I've spent far too much of the organization's money on IT equipment while working as a purchasing clerk. These jobs were mostly fun - but I never enjoy my day job more than when I put systems under the hacking microscope and look for vulnerabilities.

Security testing is arguably the most creative job in IT. Sure - there are always protocols to follow and managers to report to, but the joy of finding solutions to problems the client never knew they had never gets old. And sometimes you get to come up with solutions that no one ever asked for in the first place (but that's a different story altogether).

It can be intimidating when you realize that you will never fully master your profession. There are simply too many possible angles of attack on any IT-related issue for anyone to get the full picture. And that's where the fun lies - there's always something new to learn, or something old to reconsider. The point I'm trying to make is that working as a security tester never gets boring.

Summary

Doing security testing is fun but difficult. And chances are that it might even seem too difficult at first since there is so much to learn. New vulnerabilities are discovered every day, websites get hacked around the clock, and the bad guys seem to be more well-funded and more determined than ever. But with a little effort put into learning the art of security testing, you'll start seeing common patterns and shortcuts in no time. And before you know it, you will be able to test pretty much any type of system for vulnerabilities in a professional way.

Another thing to keep in mind is that the sooner you accept that you'll never fully understand every single aspect of security, the better. The harsh truth is that no one does. Modern-day computer systems are simply far too complex for any single person to figure out on their own. My advice is to leave the image of the solitary hacker who can break her way into anything to Hollywood, and embrace the fact that we all need a little help from time to time.

So right now is a good time to put down this book and set out to become the best security tester you can possibly be. Off you go, and good luck.